A FRACTAL IMAGE COMPRESSION ALGORITHM BASED ON IMPROVED IMPERIALIST COMPETITIVE ALGORITHM

ALI NODEHI

UNIVERSITI TEKNOLOGI MALAYSIA

A FRACTAL IMAGE COMPRESSION ALGORITHM BASED ON IMPROVED IMPERIALIST COMPETITIVE ALGORITHM

ALI NODEHI

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

SEPTEMBER 2013

Dedicated to my family that with their support and sufferance this research completed.

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to my Parents, who have always supported me emotionally and financially and dealt with difficulties while I was away for my studies. I would also thank Behoush, my beloved wife, who helped me and encouraged me for my studies.

My sincere thanks to my supervisor, Prof. Dr. Ghazali Bin Sulong for friendship, motivation, and encouragement. I also appreciate his quick and logical response to my academic problems and helpful guidelines and recommendations.

I would like to thank the staff of Universiti Teknologi Malaysia, and especially the Faculty of Computing, for their kind cooperation.

Lastly, I wish to thank all the people who have helped me during conducting my PhD project and the ones who provided such a great academic environment for research and education.

ABSTRACT

Fractal image compression (FIC) is a lossy compression method that has the potential to improve the performance of image transmission and image storage and provide security against illicit monitoring. The important features of FIC are high compression ratio and high resolution of decompressed images but the main problem of FIC is the computational complexity of the algorithm. Besides that, the FIC also suffers from a high number of Mean Square Error (MSE) computations for the best matching search between range blocks and domain blocks, which limits the algorithm. In this thesis, two approaches are proposed. Firstly, a new algorithm based on Imperialist competitive algorithm (ICA) is introduced. This is followed by a two-tier algorithm as the second approach to improve further the performance of the algorithm and reduce the MSE computation of FIC. In the first tier, based on edge property, all the range and domain blocks are classified using Discrete Cosine Transform. In the second tier, ICA is used according to the classified blocks. In the ICA, the solution is divided into two groups known as developed and undeveloped countries to maintain the quality of the retrieved image and accelerate the algorithm operation. The MSE value is only calculated for the developed countries. Experimental results show that the proposed algorithm performed better than Genetic algorithms (GAs) and Full-search algorithm in terms of MSE computation. Moreover, in terms of Peak Signal-to-Noise Ratio, the approaches produced high quality decompressed image which is better than that of the GAs.

ABSTRAK

Pemampatan Imej Fraktal (PIF) adalah satu kaedah pemampatan separa-mampat yang mempunyai potensi untuk meningkatkan prestasi penghantaran dan penyimpanan imej serta menyediakan keselamatan terhadap pemantauan haram. Ciri-ciri penting PIF adalah mempunyai nisbah pemampatan dan resolusi yang tinggi bagi imej nyahmampat tetapi algoritma pengiraannya sangat kompleks dan ia merupakan masaalah utamanya. Disamping itu, PIF juga memerlukan bilangan pengiraan Ralat Kuasa dua Terkecil (RKT) yang tinggi bagi membuat carian terbaik antara blok-perbagai dan blok-domain, justeru menghadkan lagi kemampuan algoritma tersebut. Dalam tesis ini, dua pendekatan dikemukakan. Pertamanya, satu pendekatan baru diperkenalkan berdasarkan kepada Algoritma Kompetitif Imperialis (AKI). Seterusnya untuk meningkatkan prestasi algoritma tersebut dan mengurangkan pengiraan RKT, satu algoritma dua peringkat dicadangkan sebagai pendekatan kedua. Dalam peringkat pertama, berdasarkan bentuk pinggir, semua blok-domain dan blok-perbagai dikelaskan menggunakan Transformasi Kosain Diskret. Dalam peringkat kedua, AKI digunakan mengikut blok yang telah diklasifikasikan. Dalam AKI, untuk mengekalkan kualiti imej nyahmampat dan mempercepatkan operasi algoritma, penyelesaian tersebut dibahagikan kepada dua kumpulan dan dikenali sebagai negara maju dan negara mundur. Seterusnya nilai RKT dikira untuk negara maju sahaja. Hasil eksperimen menunjukkan bahawa algoritma yang dicadangkan mempunyai prestasi yang lebih baik daripada Algoritma Genetik (AG) dan Algoritma Carian Penuh dari segi kiraan RKT. Keputusan juga menunjukkan bahawa Nisbah Puncak-Signal terhadap Hingar dan kualiti imej nyahmampat yang diperolehi oleh kaedah yang dicadangkan adalah lebih baik daripada kaedah AG.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BTC	–	Block Truncated Coding
CMFB	–	Cosine-Modulated Filter Banks
DCT	–	Discrete Cosine Transform
EA	–	Evolutionary Algorithm
EO	–	Evolutionary Optimization
FFT	–	Fast Fourier Transform
FIC	–	Fractal Image Compression
GA	–	Genetic Algorithm
HFIC	–	Huber Fractal Image Compression
ICA	–	Imperialist Competitive Algorithm
IFS	–	Iterated Function System
JPEG	–	Joint Photographic Experts Group
JPG	–	Joint Photographic Experts Group
LIFSM	–	Local Iteration Function System with gray level Maps
LIFSMD	–	Local Iteration Function System with gray level Maps and Discrete cosine transform
LIFSMI	–	Local Iteration Function System with gray level Maps and dimensional Interpolation
MA	–	Memetic Algorithm
MRCM	–	Multiple Reduction Copying Machine
MSE	–	Mean Squared Error
PIFS	–	Partitioned Iterated Function Systems
PSNR	–	Peak Signal-to-Noise Ratio
PSO	–	Participle Swarm Optimization
RBFC	–	Region-Based Fractal Coder
SNR	–	Signal-to-Noise Ratio
SGA	–	Schema Genetic Algorithm

VQ	–	Vector Quantization

LIST OF SYMBOLS

$AFDCT$	–	DCT absolute coefficient pair
β	–	A number greater than one
c_n	–	The cost of the n^{th} imperialist
C_n	–	The normalized cost of the n^{th} imperialist
CN	–	Number of all counties
d	–	The distance between colony and the imperialist position
D	–	The domain pool
T	–	Number of decades
DS	–	The diagonal/sub-diagonal edge class
f	–	A given 256×256 gray level image
$F_{(0,1)}$	–	The intensity variation between the lower half and upper half of image block
$F_{(1,0)}$	–	The intensity variation between the right half and left half of image block
HV	–	The horizontal/vertical edge class
k	–	The orientation type
λ	–	A random number with a uniform distribution in the range of $(0,1)$
$(\lvert F_{(1,0)}\rvert_N, \lvert F_{(0,1)}\rvert_N)$	–	Normalize absolute coefficient pairs
NC_n	–	The number of colonies of the n^{th} imperialist
N_{col}	–	The number of all colonies of imperialist
NE	–	number of empires
NIB	–	Number of image blocks
NTC_n	–	The normalized total cost of the n^{th} empire
p	–	The contrast of an image block
P_n	–	The imperialist's normalized power
$P_{p_{imp}}$	–	The empire's possession probability

p_t	–	The transformation type
p_x	–	The domain block's in the horizontal position
p_y	–	The domain block's in the vertical position
q	–	The brightness of an image block
R	–	The range pool
RN	–	Number of range blocks
S	–	The smooth type class
TC_n	–	The total cost of the n^{th} empire
T_D	–	The distance threshold
T_A	–	The angle threshold
u	–	A sub-sample domain block
v	–	A sub-sample rang block
(x_{old}, y_{old})	–	The old position of the country
(x', y')	–	The new position of the country after moving
(x_i, y_i)	–	The position of empire corresponding to the country
ζ	–	The positive number considered to be between 0 and 1

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

In recent years, multimedia has wieldy emerged, therefore, images are increasingly required to be stored in large number and high quality. One of the problems is that images with high quality have generally a large size. To solve this problem, it is needed to find some suitable methods for compression of images. In compression process, the amount of data needed for representation of a certain quantity of information is reduced. In other words, the compression technique aimed at eliminating the redundancy in data in such a way that a quality-acceptable image could be reconstructed. Compression of images can be useful in saving the storage space, and the compressed images need less time to be transmitted via modem. Therefore, in both cases it is completely economical. Compression algorithms also provide a level of security against illicit monitoring. Compression algorithms tend to encounter several problems such as the speed of operation and the required degree of compression that should be considered while designing new algorithms. It is obvious that in cases a program is attempted to be run directly from the compressed state of algorithms, it is important to enhance the speed of decompression process.

Compression methods can be essentially divided into two groups (Gonzalez and Woods, 2006). First, ”lossless” compression that works through reducing the redundant data. What would be decompressed is a copy exactly taken from the original, without any loss of the data. Two examples of this type of algorithms are Arithmetic Coding and Huffman Encoding. Second, ”lossy” compression in which the exact reproduction of data is ignored, instead, it attempts to provide a better compression.

It eliminates the redundant data and attempts to create something nearly identical to the original. One obvious matter is that the lossless compression methods should be applied to text files or programs, whereas the lossy methods are suitable only for sound data or graphics, in which there is no need for exact reproduction. One method based on lossy compression is fractal image compression, hereinafter called FIC (see Figure 1.1).

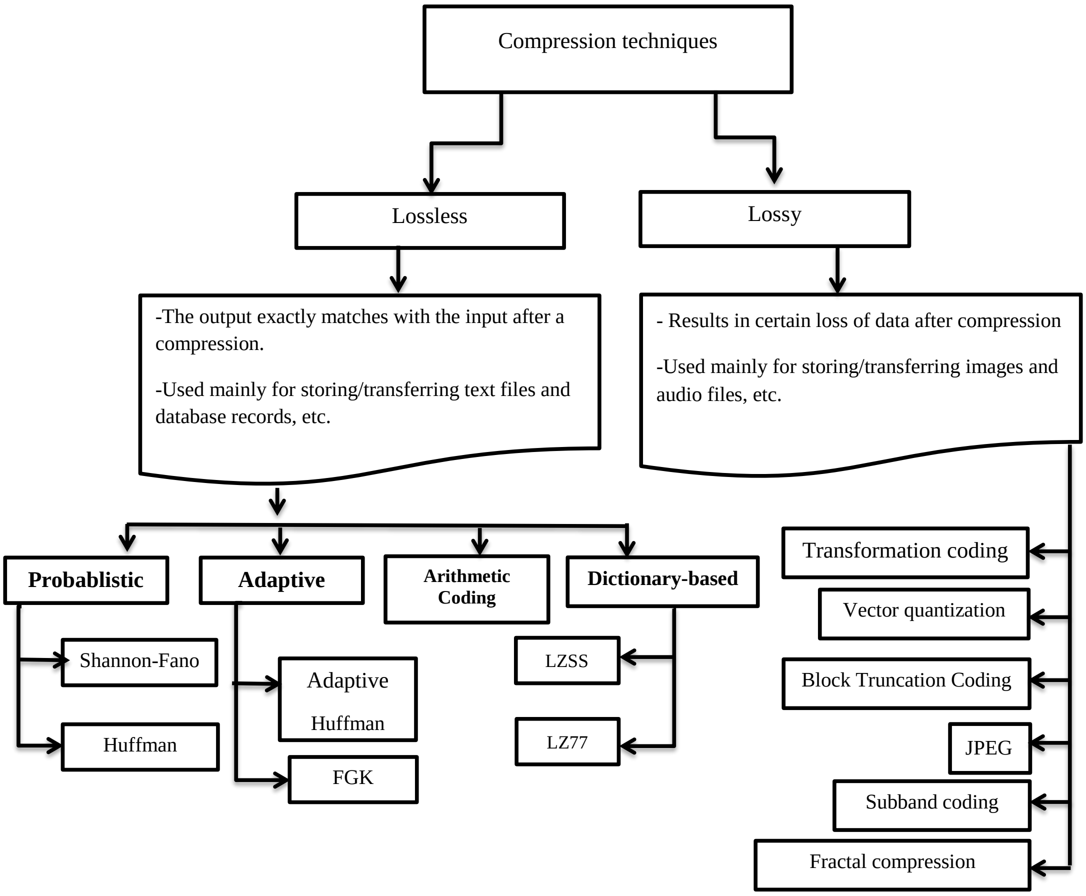

Figure 1.1: The compression techniques.

In 1975, the term 'fractal' for the first time was used by Benoit B. Mandelbrot who was a French mathematician. This word was derived from the Latin 'fractus' that meant 'fragmented and irregular' or 'broken'. Indeed, it can be said that the fractal geometry dates back to Mandelbrot and his book The Fractal Geometry of Nature (Mandelbrot, 1983). Mandelbrot, in his book, argued that the classical Euclidean geometry did not have adequate capacity for the description of numerous natural objects, such as trees, clouds, coastlines, and mountains. Therefore, he proposed and developed the fractal geometry.

With classical Euclidean shapes only two parameters, length and position, are needed to describe the shape; whereas, fractals require three parameters: complicated structure on a wide range of scales, repetition of structures at different length scales (self-similarity), and a 'fractal dimension' that is not an integer (Mandelbrot, 1983). Self-similarity is found in sets or shapes that have repetitive patterns on smaller scales. A line and a square are two Euclidean shapes that are self-similar. Enlarging and replicating the line can produce the square, just as reducing the square can form the line (Benot Mandelbrot and Cot, 1977). In this case, the line has dimension 1, and the square has dimension 2. It seems possible that a form between a line and a square, say a jagged line, can have a dimension between 1 and 2 (Benot Mandelbrot and Cot, 1977).

Fractals are categorized mainly into two groups: nonlinear and linear. Nonlinear fractals, in turn, are divided into two popular types: Julia sets and Mandelbrot set that are fractals of the complex plane (Benot Mandelbrot and Cot, 1977; Mandelbrot, 1983). Since Mandlebrot's success in making the research of fractals and their applications popular, many people have learned to create fractal illustrations. Today these beautiful images can be generated easily on a personal computer, and have spawned a popular field of computer graphics art. Some of fractal images are Sierpinski triangle, Mandelbrot set, Cantor set, Julia set, Koch curve, Apollonian gasket, Douady rabbit, Sierpinski Hexagon, Pinwheel fractal, and etc. Nevertheless, those fractals that are applied to image compression are from linear group and they are of the real plane. Therefore, the employed fractals are not chaotic; that is, they do not have any sensitivity to the initial conditions. These fractals are from Iterated Function Theory. Simply, Iterated Function System (IFS) refers to a set of contractive affine transformations. The next section, Fractal image compression is explained in detail.

1.1 Background of the Problem

The history of FIC dates from 1977 (the year Mandelbrot's book, namely, *The Fractal Geometry of Nature* was published). In 1981, Hutchinson demonstrated a branch of mathematics known as Iterated Function Theory (Hutchinson, 1981; Mandelbrot, 2004a). About one decade later, Michael Barnsley, a researcher from Georgia Tech, published a book, tittled, "*Fractals Everywhere*", which presented the mathematics of the Iterated Functions Systems (IFSs), and proved a result that was recognized as the Collage Theorem (Barnsley and Rising, 2000). This theorem determined how IFS could represent an image. It led to exploring new possibilities. If, in forward direction, the fractal mathematics creates natural looking images, then, in the reverse direction, can it not compress images? To go from a particular image to an IFS, which can create the original (or closely similar to it) is referred to as inverse problem, which so far, it has been remained unsolved (Barnsley and Hurd, 1993).

Barnsley claimed that he had solved the inverse problem with his Collage Theorem. He was granted a software patent and found the Iterated Systems Incorporated. Barnsley publicized his success in BYTE magazine (issued in January 1988). In his article, the inverse problem was not addressed; however, it presented a number of images that were compressed purportedly in excess of 10,000:1. Since that time, scholars have attempted to automate this process, but all attempts have failed. Barnsley (1988) admitted that each complex image needs nearly 100 hours for encoding and 30 minutes required for decoding on the Masscomp (Barnsley and Sloan, 1988). In March 1988, based on Barnsley's findings, one of his PhD students designed a modified scheme to represent the images, which was named Partitioned Iterated Function Systems (PIFS). Barnsley was again granted patent on an algorithm that was capable of automatically converting an image to a PIFS, compressing the image into the process. Jacquin employed the algorithm in a software (Jacquin, 1993). That was not a sophisticated and speedy algorithm, but a full automatic. It came at the price: gone was promise of 10,000:1 compression. Typically, a 24-bit color image could be compressed from 8:1 to 50:1; at the same time, it was looking "pretty good". Nevertheless, the whole current programs of FIC are designed based on the Jacquin's paper. Finally, a short story of FIC is summarized in Table 1.1.

Table 1.1: The chronology of FIC.

Year	Tittle	Type	Descriptions
1977	The Fractal Geometry of Nature	book	Benoit Mandelbrot finishes the first edition of book.
1981	Fractals and Self Similarity	book	John E. Hutchinson publishes this book.
1983	The Fractal Geometry of Nature	book	Revised edition of the book is published
1985	Iterated Function Systems and the Global Construction of Fractals	paper	Michael F. Barnsley and Stephen Demko introduce Iterated Function Theory in their paper
1987	Iterated Systems Incorporated	paper	Iterated Systems Incorporated is founded in Georgia
1988	A Better Way to Compress Images	Paper	Michael Barnsley and Alan D. Sloan wrote the article
1988	Fractals Everywhere	book	Barnsley publishes the book
1990	first patent of Barnsley's	patent	Barnsley's first patent
1991	Method and Apparatus for Processing Digital Data	Patent	M. Barnsley and A. Sloan are granted US Patent
1991	Image Data Compression Using Fractal Techniques	Paper	J.M. Beaumont publishes a paper
1992	describes the first practical fractal image compression method	Paper	Arnaud E. Jacquin publishes an article
1992	Microsoft Encarta	software	Microsoft Corporation publish the multimedia encyclopedia which uses fractal image compression to great effect
1993	Fractal Image Compression	book	Michael Barnsley and Lyman P. Hurd published this book

For compress an original image, the FIC algorithm can be described as follows. First, algorithm considers two copies of the original image. Image 1 is partitioned into non-overlapped blocks with the size of $n \times n$, each of which is known as range block. The set of these range blocks is known as range pool. Image 2 is partitioned into overlapped blocks with the size of $2n \times 2n$, each of which is known as domain block. The set of these domain blocks is known as domain pool. Figure 1.2 shows the original image and its range pool and domain pool. In FIC, for each range block, the algorithm should find a domain block that is most similar to that range block. The similarity between the candidate-range-block v and candidate-domain-block u is measured by computing the quantity d, in which p and q can be obtained as follow:

$$p = \frac{[n^2 \langle u,v \rangle - \langle u,1 \rangle \langle v,1 \rangle]}{[n^2 \langle u,u \rangle - \langle u,1 \rangle^2]} \tag{1.1}$$

and

$$q = \frac{1}{n^2}[\langle v,1 \rangle - p \langle u,1 \rangle] \tag{1.2}$$

where n is size of candidate-range-block.

$$d = \|p.u_k + q, v\|, \quad 1 \leq k \leq 8 \tag{1.3}$$

where u_k represents the eight transformations of u, and p and q are the adjustment of the contrast and brightness, respectively.

Based on this quantity, all researchers used MSE for measuring the similarity between the candidate-range-block and candidate-domain-block. The less MSE value, the more similar they will be to each other. In FIC, eight rotations are performed on each domain block, each of which is known as affine transformation. These affine transformations include rotation 0°, 90°, 180°, 270°, horizontal flip, vertical flip, flip relative to 135°, and flip relative to 45°. To find a domain block that is most similar to each candidate-range-block, each of the eight affine transformations is performed on each domain block. In each case, the MSE value is computed. The domain block with the smallest MSE value is determined as the most similar one to its candidate-range-block. For this candidate-range-block, algorithm saves the address of its corresponding domain block, the affine transformation performed on this domain block, contrast, and brightness of this block. This process is done for all range blocks.

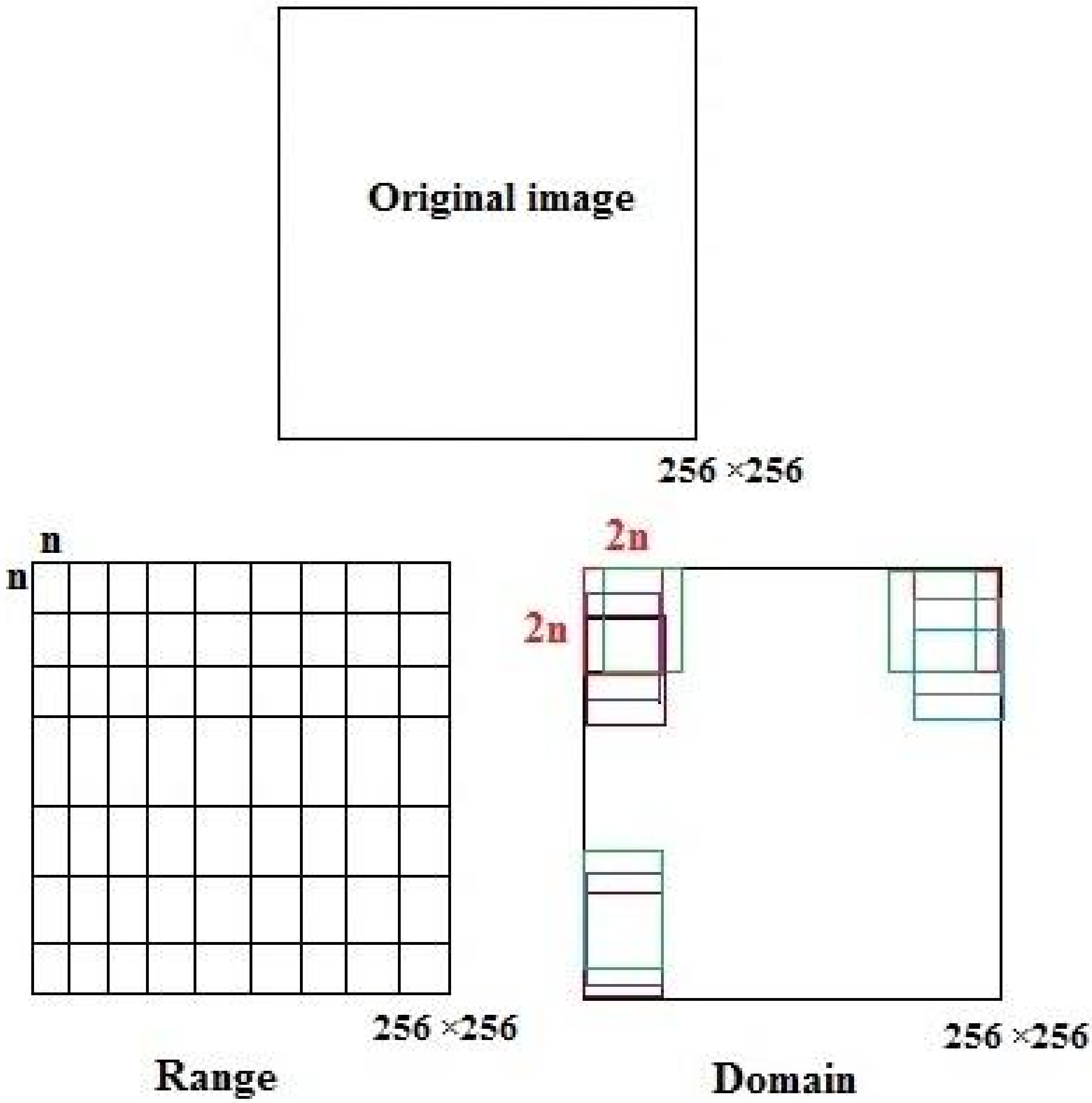

Figure 1.2: The original image and its range pool and domain pool.

For example, let f be a given 256×256 gray level image and n be 4. The range pool RP is considered as the set of all non-overlapping blocks of size 4×4, making up $(256 \div 4)^2 = 4,096$ blocks (See Figure 1.3). DP is the domain pool and it determines the set of all possible overlapping blocks of size 8×8 of the image f, making up $(256 - 8 + 1)^2 = 62,001$ blocks (See Figure 1.4). Addition of eight isometric symmetries increases this total number to 496,008 $(62,001 \times 8 = 496,008)$.

Finally, for encoding this image based on FIC method (Full-search algorithm), the $496,008 \times 4,096 = 2,031,648,768$ MSE computations are needed. With this huge amount of computations, even if a high performance workstation is used, the Full-search algorithm is very slow. Decreasing the number of MSE computations is the main challenge that FIC face.

This problem has been attractive to many researchers in this filed of study. Generally, the methods that deal with the FIC problem are divided into two groups: methods that are based on evolutionary algorithm (EA) and those that are not based on EA. Those groups are investigated as follows.

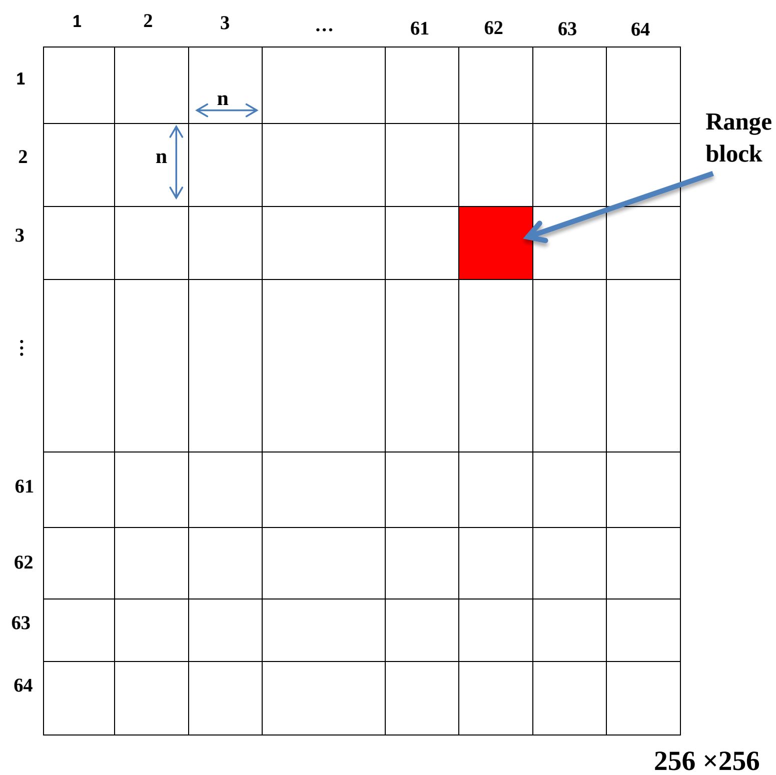

Figure 1.3: The range pool.

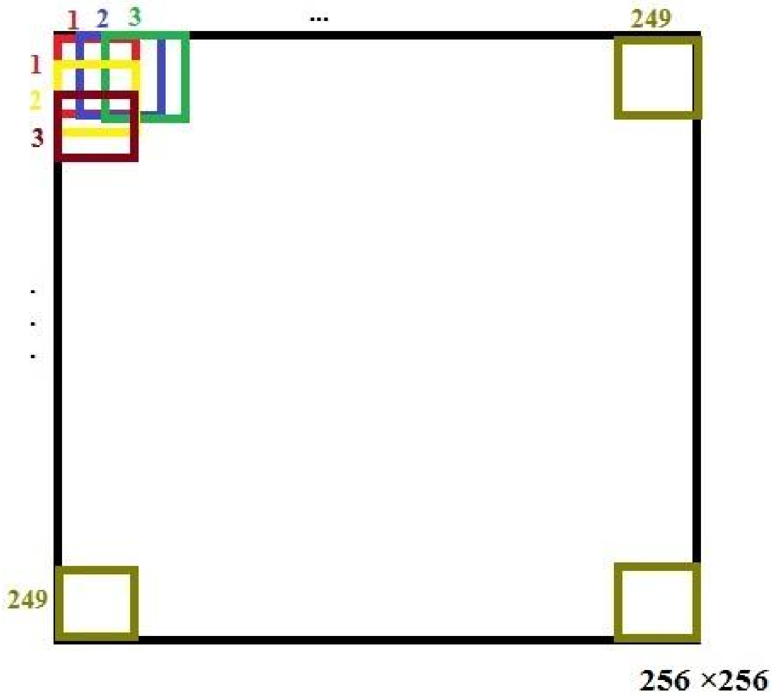

Figure 1.4: The domain pool.

During the past decades, researchers have proposed various computer simulation of natural processes for solving optimization problems. The Evolutionary Optimization (EO) is a popular and useful field of research and developing algorithms to solve many problems (Sarker *et al.*, 2002). Evolutionary algorithms (EAs) are search methods that take their inspiration from natural selection and survival of the fittest in the biological world. EAs differ from more traditional optimization techniques in that they involve a search from a "population" of solutions, not from a single point. For instance, genetic algorithm (GA) is considered as an evolutionary algorithm in which some candidate solutions are evolved for the given problems (Sivanandam and Deepa, 2010). The evolutionary algorithms can help the FIC problem by decreasing the number of MSE computations and find near optimal solutions. In several works, GA has been used to improve the FIC algorithm (Xuan and Dequn, 1996; Vences and Rudomin, 1997; Mitra *et al.*, 1998; Xun and Zhongqiu, 2000; Gafour *et al.*, 2003; Mohamed and Aoued, 2005; Liangbin and Lifeng, 2007; Wu *et al.*, 2006, 2007; Wu and Lin, 2010). In these studies, some searching strategies have been proposed by employing GA, wherein each chromosome contains the PIFS of the range blocks. In their algorithms, the chromosome fitness that is achieved by evolutionary process is measured by the distance existing between the range blocks and domain blocks. Xuan and Dequn (1996) proposed a method for finding the IFS code of fractal image, and examining the effect of mutation and crossover on FIC. Vences and Rudomin (1997) proposed a GA method for FIC. However their algorithm has a high speed, the retrieved image does not have enough quality. In studies conducted by Mitra et al. (1998) and Xun and Zhongqiu (2000), FIC is performed through GA method which is based on an elitism model. Elitism is a mechanism in GA based on which some suitable solutions are selected and transferred to the next generation. A decrease in search space is the result of finding the best self-similarity. An increase in the speed of FIC algorithm was observed despite a decrease in the quality of retrieved image. A spatial correlation genetic is proposed by Wu et al. (2006). Their algorithm consists of two stages. The first one is performing the spatial correlation in image for both domain pool and rang pool. This is performed for exploiting local optima. If the local optima obtained from the first stage is not satisfactory, the second stage will be done to find the best self-similarity in the whole image. In their study, the number of MSE computations was reduced, however, it was still too much. Wu et al. (2007) proposed a schema genetic algorithm (SGA) for FIC. In this algorithm, they adapted the genetic operators based on the schema theorem within the evolutionary process. In terms of the number of MSE computations and the quality of retrieved image, this algorithm performed better than the previous ones. A GA algorithm with a hybrid select mechanism were presented by Wu et al. (2010). In their algorithm, using elitism and selecting suitable solutions, GA could reduce the number of MSE computations. In above-mentioned studies, GA could reduce to some extend the number of MSE computations; however, one of the GA drawbacks was that it might be trapped in local optima, which led to reducing the quality of retrieved image and extra computation.

Another method that do not use EA is no-search method. Furao and Hasegawa (2004) introduced a no-search fractal encoding algorithm by which they could select a domain block with the same center as the range block; in this condition, the domain block could be encoded as a matched block. Since in a natural image, both the domain blocks and the range blocks have the same texture, and matching between the domain blocks and range block is achievable. However in this algorithm, the number of MSE computations was reduced, the quality of retrieved image was reduced, too. This is because a suitable domain block is not necessarily located near to the range block location. Wang and Wang (2008) enhanced a no-search FIC algorithm by a modified gray-level transform. The gray-level transformation, as one form of digital image processing for the reduction of redundant shadows of image. For enhancing the possibility of successful matching for a domain block and a range block, a modified gray-level transform was introduced. Afterward, a no-search FIC coding method was suggested by means of two gray-level transforms, one for the small blocks and the other for the large blocks according to the quad-tree partition scheme. It was done in order to accelerate the encoding process time, but the quality of retrieved image is not good. In another study, based on a modified gray-level transform, Wang et al. (2010) enhanced a no-search FIC coding method. To decrease the minimum matching error between a certain range block and its related domain block, they enhanced the gray-level transform and called it fitting plane. Although the fitting plane is capable of enhancing the probability of successful range-domain matching, the quality of retrieved image is not good.

However the problem of FIC has received a great attention from the research community, this problem should be investigated more. Some optimization methods could reduce to some extent the number of MSE computations; although, the quality of retrieved image has been remarkably decreased. On the other hand, some researchers have put an emphasis on reserving the quality of retrieved image, which has led to an increase in the number of MSE computations.

1.2 Statement of the Problem

Some important features of FIC are high compression ratio and high-resolution reconstructed images. On the other hand, as we described in the previous section, for each range block, the FIC algorithm should find a domain block that is most similar to that range block. To find a domain block that is most similar to each candidate-range-block, each of the eight affine transformations is performed on each domain block. In each case, the MSE value is computed. The domain block with the smallest MSE value is determined as the most similar one to its candidate-range-block. Because of huge amount of MSE computations, the basic algorithm of FIC runs very slowly, even with a high performance workstation. Therefore, the main problem of FIC is high number of MSE computations. During recent years, several methods have been designed for solving this problem. The FIC is a rough problem that has many local optima. The problem of some evolutionary algorithms is that they may be trapped in local optima. Evolutionary algorithms are considered suitable for FIC since they can reduce the number of MSE computations of FIC; however, it is a challenging issue to find an evolutionary algorithm capable of escaping from the local optima. In the FIC algorithm, for each candidate range block, a suitable domain block should be selected from among domain block pool. Another problem in FIC algorithm is how to search a domain block that is suitable for candidate range block. In the FIC algorithm, for each candidate range block, all domain blocks are computed, then the best domain block is selected as the best matched domain block with the range block. Therefore, the best quality of retrieved image is obtained. As evolutionary algorithms search for just some of the possible solutions, algorithm may fail to select the best domain block. As a result, the quality of retrieved image may reduce. Therefor, a problem is to decrease the number of MSE computations and, at the same time, to preserve the quality of retrieved image as much as possible. Another challenging issue is that not all elements of domain pool are suitable for candidate range block. A suitable strategy is to differentiate the proper and improper domain blocks, then the most suitable one can be selected from among proper domain blocks. It can decrease the search space, which can lead to acceleration of the FIC algorithm. Therefore, it is important to find a method to do appropriately this strategy.

1.3 Questions of Research

Based on research statement, there is one main question that is how to decrease the number of MSE computation of FIC and preserve the quality of retrieved image as much as possible? Based on this question, some sub questions are arisen as follow:

(i) How to enhance an evolutionary algorithm in order to improve search space?

(ii) How to eliminate ineffectual elements from domain pool?

(iii) How to increase the quality of retrieved image while decreasing the computation?

1.4 Aim of the Research

To decrease the number of MSE computations of FIC and to preserve the quality of retrieved image as much as possible by using evolutionary algorithm.

1.5 Objectives of Research

To attain research aim, the following research objectives have been identified:

(i) To develop an approach for decreasing the number of MSE computations of Fractal image compression based on Imperialist Competitive Algorithm.

(ii) To eliminate ineffectual element of domain pool and decrease the search space.

(iii) To find appropriate domain blocks and preserve the quality of retrieved image as much as possible.

1.6 Research Scope

For this study the following constraints are considered:

(i) This research concentrates on gray scale images only.

(ii) Using evolutionary algorithm, this research mainly focuses on decreasing the number of MSE computations and preserve the quality of retrieved image as much as possible.

(iii) This research doesn't focus on compression ratio.

(iv) Performance of the algorithm is evaluated and validated by MATLAB.

1.7 Research Significance

In this research, the problem of FIC is introduced. It can be a significant research because it proposes and develop several algorithms that could be used as a benchmark for newly designed algorithms and it provides a criterion for other researchers to evaluate their algorithms with a near-optimal solution. Additionally, in the present study, some efficient techniques are designed to manage the domain pool and candidate solutions. These techniques can result in decreasing the number of MSE computations of FIC and preserve the quality of retrieved image.

1.8 Organization of the Thesis

This thesis encompasses six chapters. Chapter 1 presents current challenges and historical background of FIC. In addition the objectives and scopes of thesis are in this chapter. Chapter 2 presents the methods of image compression, fractals and FIC. In this chapter the basic algorithm and mathematical background of FIC is introduced. Additionally, the related works about FIC are reviewed. In final part of Chapter 2, the Evolutionary Algorithms (EA) is presented. Chapter 3 presents the methodology of the current study. Framework and overview of the research methodology, instrumentation and data set, and evaluation metrics of the research are introduced in this chapter. Chapter 4 presents the new approaches of FIC based on ICA that led to improve performance of FIC. Chapter 5 presents the speedup FIC by ICA based on discrete cosine transform (DCT). Finally, chapter 6 concludes the achievements in this study and future works are recommended.

REFERENCES

Atashpaz-Gargari, E. and Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In *IEEE Congress on Evolutionary Computation 2007*. 25-28 September. Singapore, 4661–4667.

Barni, M. (2006). *Document and Image Compression*. USA: Taylor and Francis.

Barnsley, M. and Rising, H. (2000). *Fractals everywhere*. The University of California, USA: Academic Press Professional.

Barnsley, M. and Sloan, A. (1988). A better way to compress images. *Byte Magazine*. 9(7), 215–223.

Barnsley, M. F. and Hurd, L. P. (1993). *Fractal image compression*. The University of Michigan, USA: A K Peters.

Belloulata, K. and Konrad, J. (2002). Fractal image compression with region-based functionality. *IEEE Transactions on Image Processing*. 11(4), 351–362.

Bratteli, O. and Jrgensen, P. E. T. (1999). *Iterated function systems and permutation representations of the Cuntz algebra*. USA: American Mathematical Society.

Chang, H. T. and Kuo, C. (2000). Iteration-free fractal image coding based on efficient domain pool design. *IEEE Transactions on Image Processing*. 9(3), 329–339.

Chung, K. L. and Hsu, C. H. (2006). Novel prediction and subblock-based algorithm for fractal image compression. *Chaos, Solitons and Fractals*. 29(1), 215–222.

Cochran, W. O., Hart, J. C. and Flynn, P. J. (1996). Fractal volume compression. *IEEE Transactions on Visualization and Computer Graphics*. 2(4), 313–322.

Crownover, R. M. (1995). *Introduction to fractals and chaos*. The University of Michigan, USA: Jones and Bartlett.

Davoine, F., Antonini, M., Chassery, J. M. and Barlaud, M. (1996). Fractal image compression based on Delaunay triangulation and vector quantization. *IEEE Transactions on Image Processing*. 5(2), 338–346.

Distasi, R., Nappi, M. and Riccio, D. (2006). A range/domain approximation error-based approach for fractal image compression. *IEEE Transactions on Image

Processing. 15(1), 89–97.

Distasi, R., Polvere, M. and Nappi, M. (1998). Split decision functions in fractal image coding. *Electronics Letters*. 34(8), 751–753.

Duh, D. J., Jeng, J. H. and Chen, S. Y. (2005). DCT based simple classification scheme for fractal image compression. *Image and Vision Computing*. 23(13), 1115–1121.

Eades, A. (1994). The image processing handbook. *Microscopy Research and Technique*. 28(1), 75–95.

Fisher, Y. (2012). *Fractal Image Compression: Theory and Application*. UK: Springer.

Furao, S. and Hasegawa, O. (2004). A fast no search fractal image coding method. *Signal Processing: Image Communication*. 19(5), 393–404.

Gafour, A., Faraoun, K. and Lehireche, A. (2003). Genetic fractal image compression. In *ACS/IEEE International Conference on Computer Systems and Applications*. 14-18 July. Tunisia, 132–138.

Gao, X., Hong, B., Luo, L. and He, Z. (1997). Fractal image coding using cosine-modulated filter banks. *IEEE Transactions on Consumer Electronics*. 43(1), 62–68.

Ghosh, S. K., Mukherjee, J. and Das, P. P. (2004). Fractal image compression: a randomized approach. *Pattern Recognition Letters*. 25(9), 1013–1024.

Gonzalez, R. C. and Woods, R. E. (2006). *Digital Image Processing (3-th ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Hamzaoui, R. (1996). Decoding algorithm for fractal image compression. *Electronics Letters*. 32(14), 1273–1274.

Hamzaoui, R. and Saupe, D. (2000). Combining fractal image compression and vector quantization. *IEEE Transactions on Image Processing*. 9(2), 197–208.

Hamzaoui, R., Saupe, D. and Hiller, M. (2001). Distortion Minimization with Fast Local Search for Fractal Image Compression. *Journal of Visual Communication and Image Representation*. 12(4), 450–468.

Hankerson, D., Harris, G. A. and Johnson, P. D. (2010). *Introduction to Information Theory and Data Compression. (2-th ed.)*. New York, USA: CRC Press company.

Hart, J. C. (1996). Fractal image compression and recurrent iterated function systems. *IEEE Computer Graphics and Applications*. 16(4), 25–33.

Hartenstein, H., Ruhl, M. and Saupe, D. (2000). Region-based fractal image compression. *IEEE Transactions on Image Processing*. 9(7), 1171–1184.

He, C., Yang, S. X. and Huang, X. (2004a). Progressive decoding method for fractal image compression. *IEEE Vision, Image and Signal Processing*. 151(3), 207–213.

He, C., Yang, S. X. and Xu, X. (2004b). Fast fractal image compression based on one-norm of normalised block. *Electronics Letters*. 40(17), 1052–1053.

He, C. J., Li, G. P. and Shen, X. N. (2007). Interpolation decoding method with variable parameters for fractal image compression. *Chaos, Solitons and Fractals*. 32(4), 1429–1439.

Hufnagl, C. and Uhl, A. (2000). Algorithms for Fractal Image Compression on Massively Parallel SIMD Arrays. *Real-Time Imaging*. 6(4), 267–281.

Hutchinson, J. (1981). Fractals and Self-Similarity. *Indiana University Mathematics Journal*. 30, 713–747.

Jackson, D. J., Mahmoud, W., Stapleton, W. A. and Gaughan, P. T. (1997). Faster fractal image compression using quadtree recomposition. *Image and Vision Computing*. 15(10), 759–767.

Jacquin, A. E. (1993). Fractal image coding: a review. *Proceedings of the IEEE*. 81(10), 1451–1465.

Jeng, J. H. and Shyu, J. R. (2000). Fractal image compression with simple classification scheme in frequency domain. *Electronics Letters*. 36(8), 716–717.

Jeng, J. H., Truong, T. K. and Sheu, J. R. (2000). Fast fractal image compression using the Hadamard transform. *IEEE Vision, Image and Signal Processing*. 147(6), 571–574.

Jeng, J. H., Tseng, C. C. and Hsieh, J. G. (2009). Study on Huber Fractal Image Compression. *IEEE Transactions on Image Processing*. 18(5), 995–1003.

Kominek, J. (1994). Image Compression: An Issue of Quality. *Technical report*, 1–26.

Kou, W. (1995). *Digital Image Compression: Algorithms and Standards*. Springer.

Lalitha, E. M. and Satish, L. (1998). Fractal image compression for classification of PD sources. *IEEE Transactions on Dielectrics and Electrical Insulation*. 5(4), 550–557.

Liangbin, Z. and Lifeng, X. (2007). A Study of Fractal Image Compression Based on an Improved Genetic Algorithm. *International Journal of Nonlinear Science*. 3(2), 116–124.

Linnell, T. A. and Deravi, F. (2004). Mapping vector accumulator: fractal domain feature for character recognition. *Electronics Letters*. 40(22), 1406–1407.

Loe, K. F., Gu, W. G. and Phua, K. H. (1997). Speed-up fractal image compression with a fuzzy classifier. *Signal Processing: Image Communication*. 10(4), 303–311.

Mandelbrot, B. B. (1977). *Fractals: form, chance and dimension*. San Francisco, USA: W.H.Freeman.

Mandelbrot, B. B. (1983). *The Fractal Geometry of Nature*. San Francisco, USA: Henry Holt and Company.

Mandelbrot, B. B. (2004). *Fractals and Chaos: The Mandelbrot Set and Beyond*. USA: Springer.

Mitra, S., Murthy, C. and Kundu, M. (1998). Technique for fractal image compression using genetic algorithm. *IEEE Transactions on Image Processing*. 7(4), 586–593.

Mohamed, F. and Aoued, B. (2005). Speeding Up Fractal Image Compression by Genetic Algorithms. *Multidimensional Systems and Signal Processing*. 16(2), 217–236.

Muruganandham, A. and Banu, R. S. D. W. (2010). Adaptive fractal image compression using PSO. *Procedia Computer Science*. 2, 338–344.

Palazzari, P., Coli, M. and Lulli, G. (1999). Massively parallel processing approach to fractal image compression with near-optimal coefficient quantization. *Journal of Systems Architecture*. 45(10), 765–779.

Polvere, M. and Nappi, M. (2000). Speed-up in fractal image coding: comparison of methods. *IEEE Transactions on Image Processing*. 9(6), 1002–1009.

Ramkumar, M. and Anand, G. V. (1997). An FFT-based technique for fast fractal image compression. *Signal Processing*. 63(3), 263–268.

Ruhl, M., Hartenstein, H. and Saupe, D. (1997). Adaptive partitionings for fractal image compression. In *Proceedings of International Conference on Image Processing*. 26-29 October. Santa Barbara, 310–313.

Russ, J. C. (2011). *The Image Processing Handbook. (6-th ed.)*. USA: Taylor and Francis.

Salomon, D. (2007). *Data Compression: The Complete Reference. (3-th ed.)*. Pub-SV: Springer.

Sarker, R., Mohammadian, M. and Yao, X. (2002). *Evolutionary Optimization*. Massachusetts, USA: Springer.

Saupe, D. (1996). A new view of fractal image compression as convolution transform coding. *IEEE Signal Processing Letters*. 3(7), 193–195.

Saupe, D. and Jacob, S. (1997). Variance-based quadtrees in fractal image compression. *Electronics Letters*. 33(1), 46–48.

Shi, Y., Gu, W., Zhang, L. and Chen, S. (1997). Some new methods to fractal image compression. *Communications in Nonlinear Science and Numerical Simulation*. 2(2), 80–85.

Sivanandam, S. N. and Deepa, S. N. (2010). *Introduction to Genetic Algorithms*. New York, USA: Springer.

Soberano, L. A. (2000). The mathematical foundation of image compression. *Honors Program. The University of North Carolina*.

Sun, K. T., Lee, S. J. and Wu, P. Y. (2001). Neural network approaches to fractal image compression and decompression. *Neurocomputing*. 41(14), 91–107.

Thomas, L. and Deravi, F. (1995). Region-based fractal image compression using heuristic search. *IEEE Transactions on Image Processing*. 4(6), 832–838.

Tong, C. S. and Pi, M. (2001). Fast fractal image encoding based on adaptive search. *IEEE Transactions on Image Processing*. 10(9), 1269–1277.

Tong, C. S. and Wong, M. (2002). Adaptive approximate nearest neighbor search for fractal image compression. *IEEE Transactions on Image Processing*. 11(6), 605–615.

Truong, T. K., Jeng, J. H., Reed, I. S., Lee, P. C. and Li, A. (2000). A fast encoding algorithm for fractal image compression using the DCT inner product. *IEEE Transactions on Image Processing*. 9(4), 529–535.

Truong, T. K., Kung, C. M., Jeng, J. H. and Hsieh, M. L. (2004). Fast fractal image compression using spatial correlation. *Chaos, Solitons and Fractals*. 22(5), 1071–1076.

Tseng, C. C., Hsieh, J. G. and Jeng, J. H. (2008). Fractal image compression using visual-based particle swarm optimization. *Image and Vision Computing*. 26(8), 1154–1162.

Vences, L. and Rudomin, I. (1997). Genetic algorithms for fractal image and image sequence compression. *Proceedings Computacion Visual*, 35–44.

Vidya, D., Parthasarathy, R., Bina, T. C. and Swaroopa, N. G. (2000). Architecture for fractal image compression. *Journal of Systems Architecture*. 46(14), 1275–1291.

Wadstromer, N. (2003). An automatization of Barnsley's algorithm for the inverse problem of iterated function systems. *IEEE Transactions on Image Processing*. 12(11), 1388–1397.

Wang, X. Y., Li, F. P. and Wang, S. G. (2009). Fractal image compression based on spatial correlation and hybrid genetic algorithm. *Journal of Visual Communication and Image Representation*. 20(8), 505–510.

Wang, X. Y. and Wang, S. G. (2008). An improved no-search fractal image coding method based on a modified gray-level transform. *Computers and Graphics*. 32(4), 445–450.

Wang, X. Y., Wang, Y. X. and Yun, J. J. (2010). An improved no-search fractal image coding method based on a fitting plane. *Image and Vision Computing*. 28(8), 1303–1308.

Wein, C. J. and Blake, I. F. (1996). On the performance of fractal compression with clustering. *IEEE Transactions on Image Processing*. 5(3), 522–526.

Wohlberg, B. and De Jager, G. (1999). A class of multiresolution stochastic models generating self-affine images. *IEEE Transactions on Signal Processing*. 47(6), 1739–1742.

Wu, M. S., Jeng, J. H. and Hsieh, J. G. (2007). Schema genetic algorithm for fractal image compression. *Engineering Applications of Artificial Intelligence*. 20(4), 531–538.

Wu, M. S. and Lin, Y. L. (2010). Genetic algorithm with a hybrid select mechanism for fractal image compression. *Digital Signal Processing*. 20(4), 1150–1161.

Wu, M. S., Teng, W. C., Jeng, J. H. and Hsieh, J. G. (2006). Spatial correlation genetic algorithm for fractal image compression. *Chaos, Solitons and Fractals*. 28(2), 497–510.

Xuan, Y. and Dequn, L. (1996). An improved genetic algorithm of solving IFS code of fractal image. In *3rd International Conference on Signal Processing*. 14-18 October. Beijing, 1405–1408.

Xun, L. and Zhongqiu, Y. (2000). The application of GA in fractal image compression. In *Proceedings of the 3rd World Congress on Intelligent Control and Automation*. 28 June. Hefei, 553–555.

Zhou, Y. M., Zhang, C. and Zhang, Z. K. (2008). Fast hybrid fractal image compression using an image feature and neural network. *Chaos, Solitons and Fractals*. 37(2), 623–631.